

Predicting the Popularity of Online Serials with Autoregressive Models

Biao Chang¹ Hengshu Zhu¹ Yong Ge² Enhong Chen^{1*} Hui Xiong³ Chang Tan¹

¹University of Science and Technology of China

{chbiao, zhs, tanchang}@mail.ustc.edu.cn cheneh@ustc.edu.cn

²University of North Carolina at Charlotte, yong.ge@uncc.edu

³Rutgers University, hxiong@rutgers.edu

ABSTRACT

Recent years have witnessed the rapid prevalence of online serials, which play an important role in our daily entertainment. A critical demand along this line is to predict the popularity of online serials, which can enable a wide range of applications, such as online advertising, and serial recommendation. However, compared with traditional online media such as user-generated content (UGC), online serials have unique characteristics of sequence dependence, release date dependence as well as unsynchronized update regularity. Therefore, the popularity prediction for online serials is a nontrivial task and still under-addressed. To this end, in this paper we present a comprehensive study for *predicting the popularity of online serials* with autoregressive models. Specifically, we first introduce a straightforward yet effective *Naive Autoregressive* (NAR) model based on the correlations of serial episodes. Furthermore, we develop a sophisticated model, namely *Transfer Autoregressive* (TAR) model, to capture the dynamic behaviors of audiences, which can achieve better prediction performance than the NAR model. Indeed, the two models can reveal the popularity generation from different perspectives. In addition, as a derivative of the TAR model, we also design a novel metric, namely *favor*, for evaluating the quality of online serials. Finally, extensive experiments on two real-world data sets clearly show that both models are effective and outperform baselines in terms of the popularity prediction for online serials. And the new metric performs better than other metrics for quality estimation.

Categories and Subject Descriptors

H.2.8 [Information Systems]: Database Applications - Data Mining

Keywords

popularity prediction; online serials; autoregressive models

*Contact Author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'14, November 3–7, 2014, Shanghai, China.

Copyright 2014 ACM 978-1-4503-2598-1/14/11 ...\$15.00.

<http://dx.doi.org/10.1145/2661829.2662055>.

1. INTRODUCTION

With the rapid development of online content-sharing Websites, more and more people would like to become online audiences in their daily entertainments. A recent trend of the over-the-top (OTT) subscription services (e.g., Hulu¹ and Youku²) is to provide online serials, such as TV shows and Webisodes, to attract more online audiences. Different from the user-generated content (UGC) and movies, online serials often have many regularly updated episodes, which will continuously capture the audiences' preference. Indeed, an interesting and practical problem for this online serial service is to predict the popularity of online serials, which can help with a wide range of applications, such as serial recommendation [21, 29], online advertising [22, 14, 29, 1, 23], user behavior analysis [1, 13, 8, 32], as well as online traffic management [7, 2, 9].

In the literature, although many efforts have been devoted to the popularity prediction for traditional online contents [1, 15, 20, 27, 30, 29, 33] and time series prediction [17, 24, 11], few of those works can be directly used to solve the problem of popularity prediction for online serials due to their unique characteristics. Specifically, online serials have strong sequence dependence and release date dependence. For example, the plot often evolves between adjacent episodes, thus two adjacent episodes may have similar audiences due to the momentum of popularity. Also episodes released on weekends or holidays may attract more audiences than those on workdays. Furthermore, since different episodes are usually released on different days, online serials have the characteristic of unsynchronized update regularity. Therefore, the popularity prediction for online serials is a nontrivial task and still under-explored.

To this end, in this paper we present a comprehensive study for *predicting the popularity of online serials* by leveraging autoregressive models. Firstly, we find that the popularity of two adjacent episodes often have a strong linear correlation. Therefore, we introduce a straightforward yet effective *Naive Autoregressive* (NAR) model based on the correlation. In this model, the popularity of a newly released episode is represented by a linear combination of the popularity of previous episodes. However, this model cannot explicitly model the dynamic behaviors of audiences, which play an important role in the popularity generation of online serials. Therefore, we further develop a sophisticated model, namely *Transfer Autoregressive* (TAR) model. In this model, we assume that audiences of an episode consist of two parts, i.e., followers who have watched previous episodes and also follow the current episode, and freshers who watch the serial for the first time such as random surfers and those who have watched the previous episodes

¹<http://www.hulu.com/>

²<http://www.youku.com/>

of this serial through other ways. Indeed, both of the two models are effective for predicting the popularity of online serials, but can reveal the popularity generation from different perspectives. In addition, as a derivative of the TAR model, we also design a novel metric, namely *favor*, for evaluating the quality of online serials. It can help to find serials of high quality, which have the high potential to attract audiences to watch.

Specifically, the contributions of this paper can be summarized as follows.

- To the best of our knowledge, this paper is the first comprehensive study for predicting the popularity of online serials with autoregressive models, which can help various novel applications, such as online advertising and recommendations.
- To solve the problem, we propose two adapted autoregressive models, namely NAR and TAR, which can reveal the popularity generation from different perspectives. Particularly, we also design a novel metric derived from the TAR model for evaluating the quality of serials.
- We conduct extensive experiments on two real-world data sets for validating the performance of the proposed models. The experimental results clearly show that both models are effective and outperform baselines in terms of the popularity prediction for online serials. And our new quality metric can lead to a better performance compared with other metrics such as average views in the first week after release.

Overview. The remainder of this paper is organized as follows. We introduce the preliminary of popularity prediction for online serials in Section 2. Section 3 and 4 discuss the details of our NAR and TAR models, respectively. In Section 5, we report the experimental results on two real-world data sets. Section 6 provides a brief review of related works. Finally, we conclude the paper and describe possible future works in Section 7.

2. PRELIMINARIES

In this section, we first introduce some preliminaries about online serials and then provide the formal problem statement of *the popularity prediction for online serials*.

Online serials include various kinds of contents, such as TV series (teleplays), Webisodes and serial novels. Let us take TV series, one of the most important components of online serials, as an example. As a convention, TV series providers always release limited (e.g., one or two) episodes at each broadcast time, and the new episodes will be released regularly (e.g., per day or per week). The episodes can be viewed by online audiences anytime after they are released. Therefore, views of released episodes at different time compose the popularity records of TV series.

Formally, each online serial s_n contains M_n episodes, i.e., $s_n = \{e_1, e_2, \dots, e_{M_n}\}$, where episode e_i is released at timestamp t_i , and $\forall i \in [1, M_n]$ we have $t_i - t_{i-1} = \Delta T$, which is the release interval between two adjacent episodes. Particularly, sometimes the online serial providers may release two or more episodes at the same time. In this case, the episodes which are released at the same time can be regarded as one episode, without loss of generality. After its release, the views intuitively constitute the popularity of an episode. Therefore, if we sample every unit time after its release time t_i , the popularity record of e_i can be represented by a time series $\phi_i = \{v_{i1}, v_{i2}, \dots\}$, where v_{ij} denotes the popularity of the episode e_i within the j -th unit time interval after release. The granularity of unit time depends on the demands of popularity modeling, which can be one day, one week,

		Time Index						
		1	2	3
Episode ID	1	v_{11}	v_{12}	v_{13}	...			
	2		v_{21}	v_{22}	...			
	3			v_{31}	...			
			
	i					?	...	?

Figure 1: The popularity records of an online serial, where v_{ij} denotes the popularity of the i -th episode within the j -th unit time interval after it is released.

and etc. We will set the unit time interval to be the release interval, ΔT , in the following description. Note that, the popularity records of different episodes contain different number of observations at a given timestamp. In other words, for s_n , the time series ϕ_i ($1 \leq i \leq M_n$) of different episodes e_i ($1 \leq i \leq M_n$) may have different lengths. Furthermore, the popularity record of s_n can be denoted as $\Phi_n = \{\phi_1, \phi_2, \dots, \phi_i\}$ ($i \leq M_n$). For example, Figure 1 demonstrates the popularity record of a specific online serial, where the update frequency of episodes is one unit time (i.e., $\Delta T = 1$).

With the above preliminaries, we can formally define the problem of popularity prediction for online serials as follows.

PROBLEM 1 (POPULARITY PREDICTION FOR ONLINE SERIALS).

Given an online serial s_n with i released episodes and its popularity record $\Phi_n = \{\phi_1, \phi_2, \dots, \phi_i\}$, the goal is to predict the popularity record of the next episode after it is released, i.e., $\phi_{i+1} = \{v_{i+1,1}, v_{i+1,2}, \dots\}$.

According to the above definition, the major task of the proposed problem is to use the popularity records of historical episodes for predicting the popularity of online serials. Indeed, the popularity of different episodes have a strong characteristic of sequence dependence, which can be captured by autoregressive modeling [17]. To be specific, we can formulate the popularity prediction task of individual serial as the following autoregressive problem:

$$\phi_{i+1} = \mathcal{M}(\phi_1, \phi_2, \dots, \phi_i), \quad (\phi_i \in \Phi_n). \quad (1)$$

Therefore, our goal is to build and learn the autoregressive model \mathcal{M} . The above formulation is not equal to the traditional autoregressive modeling problem, which only takes consideration of the historical values of a given variable (e.g., ϕ_{i+1}). However, since all the popularity records (e.g., ϕ_i) belong to the same serial s_n , the formulation can be regarded as leveraging historical records of s_n for predicting the unknown records of s_n . Therefore, our formulation still satisfies the problem definition of autoregressive modeling. In the following sections, we will introduce two adapted autoregressive models for solving this problem.

3. NAIVE AUTOREGRESSIVE MODEL

In this section, we introduce the first novel autoregressive model, namely the *Naive Autoregressive* (NAR) model.

Intuitively, if people like the first several episodes of a serial, they are likely to continue viewing the next episodes. Therefore, the major audiences of different episodes of a serial are same, and usually have personalized but relatively stable watching behaviors. For example, some big fans always watch the new episodes at the first day after they are released, while some others may watch them later on weekends due to their work or study. Indeed, according to

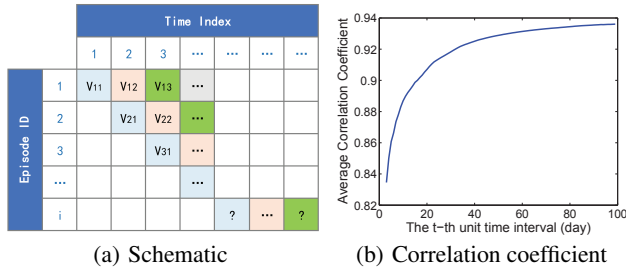


Figure 2: (a) Schematic of the NAR model; (b) The average Pearson correlation coefficient of the popularity of two adjacent episodes in our data set, which is calculated from the release time to the t -th unit time interval.

some previous reports [31, 16], such personalized watching behaviors are especially significant for online serials. Inspired by this, we propose the NAR model by utilizing the popularity of the already released episodes to predict the popularity of a new episode within the same j -th unit time interval after it is released. For example, as shown in Figure 2(a), we can use the popularity with the same color (e.g., $v_{11}, v_{21}, \dots, v_{i-1,1}$) to predict the popularity of the newly released episode (e.g., v_{i1}).

To further validate the model, we compute the average Pearson correlation coefficient between the popularity records of two adjacent episodes (i.e., ϕ_i and ϕ_{i+1}) with one real-world data set. The results are shown in Figure 2(b). We can observe that there is a strong linear correlation between the popularity of two adjacent episodes, and the average correlation coefficient becomes larger as we consider longer duration. Formally, the popularity of an episode within the j -th unit time interval (i.e., v_{ij}) is a linear combination of the popularity of previous r episodes ($e.g., v_{i-1,j}, v_{i-2,j}, \dots, v_{i-r,j}$). Specifically, we can predict v_{ij} as

$$v_{ij} = \omega_0 + \omega_1 v_{i-1,j} + \dots + \omega_r v_{i-r,j} = \mathbf{w}^T \mathbf{V}, \quad (2)$$

where $\mathbf{w} = (\omega_0, \omega_1, \dots, \omega_r)^T$ is the model parameters to be learned and $\mathbf{V} = (1, v_{i-1,j}, \dots, v_{i-r,j})$.

Particularly, to achieve better prediction performance, here we propose two different kinds of methods for learning the model parameters, which can fit our model into different application cases.

3.1 NAR with Sharing Parameters

This method indicates that all online serials should have the same model parameters. Specifically, given a training set of online serials S and their popularity records $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_{|S|}\}$, the process of learning model parameters can be implemented by minimizing the Mean Relative Squared Error (MRSE) [21, 20] loss function; that is

$$\arg \min_{\mathbf{w}} \{L(\mathbf{w}) = \frac{1}{|\Phi|} \sum_{\Phi_n \in \Phi} \sum_{v_{ij} \in \Phi_n} (\frac{\mathbf{w}^T \mathbf{V}}{v_{ij}} - 1)^2\}, \quad (3)$$

where v_{ij} is the real popularity. Here, we can transform this ordinary least squares problem to the normal equation by setting the derivative of Equation (3) with respect to \mathbf{w} to zero and solving for \mathbf{w} , and then get the optimal solution $\hat{\mathbf{w}}$ [19].

3.2 NAR with Individual Parameters

This method assumes that every serial has its own model parameters, which can capture its own popularity trend. However, a critical challenge along this line is that some serials may only have few released episodes as training data, which may increase the risk of over-fitting during model training [4]. To this end, we add a penalty term to the loss function (i.e., Equation 3) in order to avoid the case

of over-fitting. Indeed, the most widely used form of such penalty term is the sum of the squares of coefficients [4]. Therefore, the loss function changes to

$$\arg \min_{\mathbf{w}} \{L(\mathbf{w}) = \frac{1}{|\Phi_n|} \sum_{v_{ij} \in \Phi_n} (\frac{\mathbf{w}^T \mathbf{V}}{v_{ij}} - 1)^2 + \lambda \|\mathbf{w}\|^2\}, \quad (4)$$

where $\|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w} = \omega_0^2 + \omega_1^2 + \dots + \omega_r^2$ and λ keeps a trade-off between the regularization term and the loss function. This optimization problem can also be solved by the method used for learning NAR with sharing parameters. Particularly, in order to set a most appropriate value for λ , we use the value which minimizes the prediction error on the validation set.

Indeed, sharing parameters can minimize the overall MRSE while individual parameters lays emphasis on individual popularity trend. Therefore, when we only know the popularity of a few episodes of the given serial, it is better to use the former method. In contrast, when there are enough released episodes of the given serial as training data, the latter method is better.

4. TRANSFER AUTOREGRESSIVE MODEL FOR POPULARITY PREDICTION

In this section, we propose a sophisticated model, namely the *Transfer Autoregressive (TAR)* model, for popularity prediction.

Although the NAR model can exploit the linear correlation between different episodes for popularity prediction, it only uses the popularity of historical episodes with the same color in Figure 2(a). With the real-world observations, we found that the popularity of a specific episode always consists of two parts. The first part is from the followers who have watched the previous episodes and also would like to follow the current episode. These followers are loyal fans of this serial and will likely keep watching the following episodes. However, some of the followers may lose their interests as time unfolds due to many reasons, such as finding another interesting serial. The second part is from the freshers who watched the serial for the first time, such as random surfers on the Web and those who have watched the previous episodes of this serial through other ways (e.g., TV and other websites). If the freshers like this serial, they will continue watching the following episodes and become followers. Particularly, instead of watching the new episode on the first day after it is released, the audiences might watch it later. The above observations are the basic concepts of our TAR model. In the following, we will state this idea in a formal way.

ASSUMPTION 1. *Audiences of an episode are composed of followers and freshers. Both of them will sequentially watch future episodes from the episode that they start watching in this serial.*

Indeed, Assumption 1 can capture the collaborative behaviors of audiences. For better illustration, here we only take serial s_n as example. Specifically, for an episode e_i , we use a_{ij} and b_{ij} to denote the popularity from followers and freshers within the j -th unit time interval after it is released, respectively. In particular, given $i = 1$, we have

$$a_{1j} = 0, \quad \forall j \geq 1. \quad (5)$$

It is because that no one has watched this serial before. Furthermore, let p_{ijk} and q_{ijk} denote the probabilities that followers and freshers of episode e_i within the j -th unit time interval after e_i is released will keep watching the next episode e_{i+1} on its k -th unit time interval after e_{i+1} is released, respectively. Particularly, the subscripts of the two probabilities should satisfy that $t_{i+1} + k > t_i + j$, where t_i and t_{i+1} are the release times of episodes e_i and

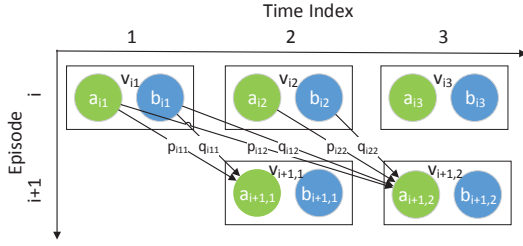


Figure 3: Graphical representation of the TAR model.

e_{i+1} , respectively. It is because that only the historical audiences will transfer to watch the current episode.

Figure 3 shows the graphical representation of the above definitions. In this figure, the two episodes e_i and e_{i+1} are released at the time index 1 and 2, respectively. Every rectangle denotes the popularity v_{ij} , which contains two circles. The green and blue circles represent a_{ij} and b_{ij} , respectively. Except for the last episode, every green and blue circles should have many weighted arrows pointing to the next episode, such as $p_{i11}, q_{i11}, p_{i12}, q_{i12}$. The weights denote the probability of which audiences will transfer from one circle to another circle, and should be larger for more popular serials. To be more specific, they satisfy the following equations:

$$\hat{v}_{ij} = a_{ij} + b_{ij} \quad (6)$$

$$a_{ij} = \sum_k (a_{i-1,k} \cdot p_{i-1,k,j}) + \sum_k (b_{i-1,k} \cdot q_{i-1,k,j}) \quad (7)$$

$$\sum_k p_{ijk} = 1, \sum_{k=j}^{\infty} q_{ijk} = 1. \quad (8)$$

Intuitively, p_{ijk} should become smaller as k increases, since most people will watch the new episode within the first unit time (e.g., the first day) after it is released. To capture this point, we further make another assumption as follows.

ASSUMPTION 2. *The number of freshers for new episodes has a trend of decreasing as more episodes are released.*

If some freshers watched an episode of the online serial, but do not like it, they would not watch the next episodes any more. This offers an intuitive explanation of Assumption 2. Another meaning of Assumption 2 is that the number of freshers, i.e., b_{ij} , is decreasing as i increases with fixed j . Indeed, for an online serial provider, the number of users is finite and the freshers for a serial usually have a maximum number. Therefore we can simplify the problem and assume that

$$b_{ij} = b_{0j} \exp^{-\alpha i}, \quad (9)$$

where α is the decay factor and b_{0j} is the initial public attention of the serial within the j -th unit time interval. Intuitively, α should be smaller while b_{0j} should be larger for popular serials because they attract more freshers as more episodes are released. After substituting Equation (5) and (9) into Equation (6) and 7, we have

$$\begin{aligned} \hat{v}_{ij} = f(m, \mathbf{b}_0, \mathbf{p}, \mathbf{q}) &= b_{0j} \cdot m^i + \sum_{k_1} b_{0,k_1} \cdot m^{i-1} \cdot q_{i-1,k_1,j} + \\ &\sum_{k_1} \sum_{k_2} b_{0,k_2} \cdot m^{i-2} \cdot q_{i-2,k_2,k_1} \cdot p_{i-1,k_1,j} + \dots + \\ &\sum_{k_1} \dots \sum_{k_{i-1}} b_{0,k_{i-1}} \cdot m \cdot q_{1,k_{i-1},k_{i-2}} \cdot p_{2,k_{i-2},k_{i-3}} \dots p_{i-1,k_1,j} \end{aligned} \quad (10)$$

where $m = \exp^{-\alpha}$, $\mathbf{b}_0 = (b_{01}, \dots, b_{0j})$, $\mathbf{p} = (p_{111}, p_{112}, \dots, p_{i-1,jj})$ and $\mathbf{q} = (q_{111}, q_{112}, \dots, q_{i-1,jj})$. Although the above equation

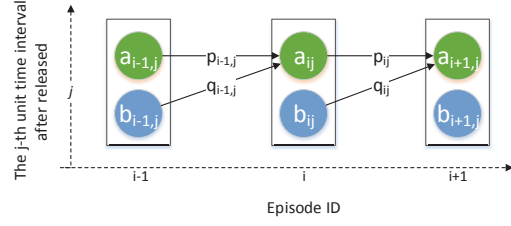


Figure 4: Graphical representation of the STAR model.

gives us an approach to predict v_{ij} , it contains so many parameters that it is hard to solve with efficient solutions. Therefore, in the following subsection, we will simplify Equation (10) by considering less transformation situations.

4.1 Single-chain Transfer Autoregressive

Indeed, the probability p_{ijk} will become smaller as k increases when fixing i and j due to the fading of popularity momentum. Moreover, as mentioned in the NAR model, the major audiences usually have personalized but relatively stable watching behaviors. Based on the above, we simplify the original TAR model by constraining that a_{ij} (or b_{ij}) can only horizontally transfer to the next episode, i.e., $a_{i+1,j}$ ($b_{i+1,j}$). Consequently, the complex TAR model in Figure 3 can be simplified to the *Single-chain Transfer Autoregressive* (STAR) model, which is shown in Figure 4.

With the STAR model, the Equation (7) can be rewritten by

$$a_{ij} = a_{i-1,j} \cdot p_{i-1,j} + b_{i-1,j} \cdot q_{i-1,j}, \quad (11)$$

where p_{ij} and q_{ij} are the probabilities that the audiences will transfer from the i -th episode to the next episode within the j -th unit time interval after they are released. Once again, by recursively substituting Equation (5), (9) and (11) into Equation (6), we can obtain

$$\begin{aligned} \hat{v}_{ij} &= a_{ij} + b_{ij} = a_{i-1,j} \cdot p_{i-1,j} + b_{i-1,j} \cdot q_{i-1,j} + b_{ij} \\ &= \dots \\ &= \sum_{n=1}^{i-1} (b_{nj} q_{nj} \prod_{k=n+1}^{i-1} p_{kj}) + b_{ij}. \end{aligned} \quad (12)$$

In particular, the above parameters can be learned by solving the following optimization problem.

PROBLEM 2 (TRANSFER OPTIMIZATION PROBLEM).

$$\min_{p_{ij}, q_{ij}, m, b_{0j}} \{L(p_{ij}, q_{ij}, m, b_{0j}) = \frac{1}{2|C|} \sum_{i \in C} (\frac{\hat{v}_{ij}}{v_{ij}} - 1)^2\}, \quad (13)$$

$$\text{subject to : } 0 < p_{ij}, q_{ij}, m \leq 1, b_{0j} > 0,$$

where \hat{v}_{ij} represents the predicted value of v_{ij} , and C is the episodes of a serial for model training.

For a given serial, p_{ij} and q_{ij} should change slightly as i increases when fixing j , because its plot evolves steadily so that the probability that people will keep following the next episodes should not have sharp changes. Therefore, to make Equation (12) more simpler, we divide p_{ij} and q_{ij} into three situations. Specifically, as i ($i \geq 1$) increases when fixing j , we have

1. Both p_{ij} and q_{ij} are constant.

2. Both p_{ij} and q_{ij} are increasing; that is

$$p_{ij} = \frac{p_{0j} \cdot i}{1 + i}, \quad q_{ij} = \frac{q_{0j} \cdot i}{1 + i}. \quad (14)$$

3. Both p_{ij} and q_{ij} are decreasing; that is

$$p_{ij} = \frac{p_{0j} \cdot (i+1)}{2i}, \quad q_{ij} = \frac{q_{0j} \cdot (i+1)}{2i}. \quad (15)$$

In the above equations, p_{0j} and q_{0j} are the initial values. Indeed, the reason why we set the values of p_{ij} and q_{ij} like Equation (14) and (15) is that those formulas can help to understand the changes of p_{ij} and q_{ij} , and thus simplify the computation of Equation (12).

To optimize the Problem 2, let us start with the first situation and rewrite Equation (12) as follows.

$$\hat{v}_{ij} = \begin{cases} q_{0j} \cdot b_{0j} \frac{(mp_{0j}^{i-1} - m^i)}{p_{0j} - m} + b_{0j} \cdot m^i & \text{if } p_{0j} \neq m \\ q_{0j} \cdot b_{0j} (i-1)p_{0j}^{i-1} + b_{0j} m^i & \text{if } p_{0j} = m \end{cases}, \quad (16)$$

where $p_{1j} = p_{2j} = \dots = p_{ij} = p_{0j}$ and $q_{1j} = q_{2j} = \dots = q_{ij} = q_{0j}$. Although the above equation is more simpler than the original Equation (12), the Problem 2 is still a non-convex optimization problem [6] and its derivatives are not easy to be calculated. Therefore, if using the classic Gradient Descent algorithm to solve the problem, we would only get a local minimum solution.

Fortunately, we find that p_{0j} , q_{0j} and m are all limited within the range (0, 1]. More importantly, when fixing other parameters (i.e., p_{0j} , q_{0j} and m), \hat{v}_{ij} is proportional to b_{0j} and the Problem 2 becomes a convex optimization problem with respect to b_{0j} . By setting the derivative of Equation (16) with respect to b_{0j} to zero and solving for b_{0j} , we can obtain

$$b_{0j} = \frac{\sum_{i \in C} \frac{d_i}{v_{ij}}}{\sum_{i \in C} \frac{d_i^2}{v_{ij}^2}}, \quad (17)$$

where we have

$$d_i = \begin{cases} q_{0j} \frac{(mp_{0j}^{i-1} - m^i)}{p_{0j} - m} + m^i & \text{if } p_{0j} \neq m \\ q_{0j} (i-1)p_{0j}^{i-1} + m^i & \text{if } p_{0j} = m. \end{cases} \quad (18)$$

To get the most appropriate values of p_{0j} , q_{0j} , m and b_{0j} , here we develop an effective algorithm named *Partial Grid Search*. Specifically, Algorithm 1 demonstrates the pseudo code of our algorithm.

The basic idea of Algorithm 1 is to fix p_{0j} , q_{0j} and m by grid search, and calculate b_{0j} according to Equation (17). Specifically, we first set the variable to be $+\infty$ in line 1. Then we use the grid search method to retrieve the value spaces of p_{0j} , q_{0j} and m . For every combination of p_{0j} , q_{0j} and m , we get b_{0j} according to Equation (17). In the innermost *for* loop from line 7 to 16, we utilize the fixed p_{0j} , q_{0j} , m and b_{0j} to predict the popularity of every episode in the training data set and accumulate the Relative Squared Error (RSE). If the summation of RSE is smaller than the given threshold $minError$, we will record the values of p_{0j} , q_{0j} , m and b_{0j} . At last, the final values of p_{0j} , q_{0j} , m and b_{0j} are those which minimize the sum of RSE.

For another two situations, it is easy to extend by using Equation (14) or (15) to rewrite Equation (12) and changing Algorithm 1 correspondingly. Particularly, by tuning $step$, which is used to set the searching step size, we can improve the precision of the final solutions of Problem 2. However, larger $step$ means higher time complexity of the algorithm. Therefore, in Section 5, we will compare the effects of different $step$ on the prediction performance.

Indeed, the transfer of audiences from one episode to another episode can intuitively explain the correlation between adjacent episodes. It also reflects the collaborative behavior of audiences.

Algorithm 1: Partial Grid Search($C, j, step$)

```

input :  $C$  (the training episode set of a serial)
input :  $j$  (views of the new episode within the  $j$ -th unit time interval after it is released)
input :  $step$  (the searching step size, e.g.,  $step = 0.01$ )
output:  $p_{0j}, q_{0j}, m, b_{0j}$  (parameters of the Transfer Model)
1  $minError = +\infty$ ;
2 for  $pp = 0; pp \leq 1; pp = pp + step$  do
3   for  $qq = 0; qq \leq 1; qq = qq + step$  do
4     for  $mm = 0; mm \leq 1; mm = mm + step$  do
5        $error = 0$ ;
6        $bb = Equation(17)$ ;
7       for  $i \in C$  do
8          $real =$  views of the  $i$ -th episode within the
            $j$ -th unit time interval after it is released;
9          $prediction = Equation(16)$ ;
10         $error += (\frac{prediction - real}{real})^2$ ;
11      if  $error < minError$  then
12         $minError = error$ ;
13         $p_{0j} = pp$ ;
14         $q_{0j} = qq$ ;
15         $m = mm$ ;
16         $b_{0j} = bb$ ;
17 return  $p_{0j}, q_{0j}, m, b_{0j}$ ;

```

4.2 Quality Estimation for Online Serials

Generally, serials of high quality will attract more audiences and obtain higher ratings. Intuitively, if the serials providers can recommend serials of high quality to audiences, they will obtain more in revenue. However, how to measure the quality of serials accurately and automatically is still an open question. So far, there are two popular and straightforward ways to evaluate the quality of serials, namely *popularity* and *rating*. The former takes the popularity (e.g., views) as a measure of quality and it assumes that more popularity indicates higher quality. While the latter allows users to rate serials that they have seen, and assumes that serials of high quality should get higher ratings. However, both of the metrics contain obvious drawbacks for quality estimation. Specifically, sometimes the popularity is not reliable, since some serials of poor quality may obtain high popularity due to the marketing promotions, such as advertising. Meanwhile, sometimes the rating is not available in some online serial Websites. Therefore, it is appealing to find some new metrics to evaluate the quality of online serials.

Particularly, we find that parameters of the TAR model, namely \mathbf{p} , \mathbf{q} , m and \mathbf{b}_0 , have the following characteristics.

- \mathbf{p} and \mathbf{q} represent the probabilities that followers and freshers like the serial, and m is the inverse of the exponential decay factor. Therefore, serials of higher quality should have larger values of \mathbf{p} , \mathbf{q} and m , and vice versa.
- \mathbf{b}_0 can be seen as the initial public attention of the serials during the corresponding unit time interval.

Therefore, we propose to combine the above parameters for designing a novel metric, namely *favor*, to evaluate the quality of serials. Specifically, the metric is defined as

$$favor \propto f(\mathbf{p}, \mathbf{q}, m, \mathbf{b}_0). \quad (19)$$

Intuitively, larger *favor* indicates higher quality and vice versa. Note that, in the single-chain TAR model, \mathbf{p} , \mathbf{q} and \mathbf{b}_0 are decided by

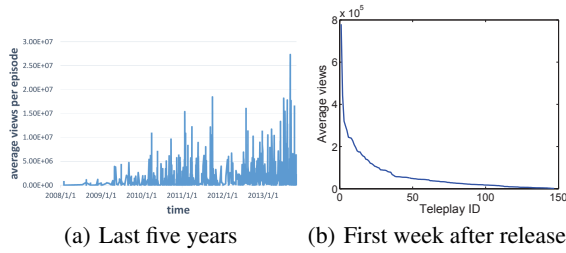


Figure 5: (a) Average views of teleplays in last five years; (b) Average views of teleplays during the first week in the Mainland sublist.

p_{0j} , q_{0j} and b_{0j} . Based on different combinations of p_{0j} , q_{0j} , m and b_{0j} , there are four straightforward possible forms of $favor$ as follows.

1. $favor_1 \propto p_{0j} \cdot q_{0j} \cdot m$.
2. $favor_2 \propto p_{0j} \cdot q_{0j} \cdot m \cdot b_{0j}$.
3. $favor_3 \propto p_{0j} + q_{0j} + m$.
4. $favor_4 \propto (p_{0j} + q_{0j} + m) \cdot b_{0j}$.

In particular, $favor_2$ and $favor_4$ consider the initial public attention of the serial while $favor_1$ and $favor_3$ are not. In the experiment section, we will empirically examine the performance of these metrics for quality estimation.

5. EXPERIMENTS

In this section, we present experimental results on two real-world data sets for evaluating the performance of our models in terms of popularity prediction and quality estimation.

5.1 Datasets

The real-world data sets used in our experiment are collected from *Teleplay List*³ of Youku, which includes teleplays of many countries or areas, e.g., Chinese Mainland, Korea and America. These teleplays are released in two different ways: 1) Youku releases all episodes of a teleplay one time after this teleplay has been updated on other sources, such as offline TV broadcast. 2) Youku updates the episodes periodically and keeps synchronous or a little later with other sources. Indeed, only teleplays released in the second way are regarded as online serials according to our definition. Some statistics of our data sets are shown in Figure 5. Specifically, Figure 5(a) demonstrates the distribution of average views of 1,930 teleplays in last five years, which clearly indicates the increasing prevalence of online TV series. Figure 5(b) shows the distribution of average views per episode in the first week after release of 147 Mainland teleplays released from May 2012 to October 2013. It indicates that there is obvious difference between the popularity of different serials, and that very few teleplays own most of the views. Thus it is needed to predict the popularity of online serials.

To guarantee the quality of experimental data, we further choose two sublists (i.e., *Mainland* and *American*) as representatives and only focus on the teleplays which were released from May 2012 to October 2013.

- **Mainland.** One or more episodes of most mainland teleplays are released every day. In other word, the update frequency of these teleplays is a day.
- **American.** One episode of most American teleplays is released every week. Therefore, the update frequency of these teleplays is a week.

³http://www.youku.com/v_olist/c_97.html

Table 1: Statistics of our data sets

Name	# of teleplays	# of episodes	Average episodes
<i>Mainland</i>	147	5,258	35.7
<i>American</i>	45	513	11.4

The detailed statistics of our two data sets are shown in Table 1. We can find that *Mainland* teleplays usually have more episodes than *American* ones. Particularly, Youku may update some teleplays just around the midnight, which results in incomplete views of the first day. Therefore, we deleted the records of the first day and treat the second day as the first day. Moreover, in the experiment, we set the unit time of *Mainland* teleplays to a day, while the unit time of *American* teleplays is set to a day and a week respectively, for evaluating the impact of different unit time with respect to popularity prediction.

5.2 Baseline Methods and Evaluation Metrics

5.2.1 Baseline Methods

To evaluate the performance of our models in terms of popularity prediction, we exploit two baselines in our experiments.

- **Average method.** It uses the average views of previous n ($n \geq 1$) episodes to predict the popularity of a new episode.
- **Vector Autoregression (VAR)** [17, 26]. Although VAR can be used to capture the linear interdependencies among multiple time series, it cannot directly be leveraged to predict the popularity of online serials. Because episodes of online serials are released at different time, which makes the popularity vector at a timestamp of different episodes contain different dimensions. Therefore, in our experiments, we extend VAR by setting the components of a vector without views to be zero, and use the extended first-order 1-VAR as a baseline method. Particularly, VAR of other orders cannot be solved due to the singularity.

5.2.2 Evaluation Metrics

We utilize the *Mean Relative Squared Error (MRSE)*, as widely used in relevant works [20, 21], to evaluate the performance of popularity prediction. Specifically, it is defined as

$$MRSE = \frac{1}{|C|} \sum_{i \in C} (\frac{\hat{v}_{ij}}{v_{ij}} - 1)^2, \quad (20)$$

where \hat{v}_{ij} is the predicted views of the real value v_{ij} , and C is the test episode set. Note that, here we use the relative error, instead of the absolute error, because views of teleplays are very different (e.g., as shown in Figure 5(b)).

Since quality estimation can be seen as a ranking problem that serials of high quality should be ranked higher, we adopt the widely used ranking measure, namely *Normalized Discount Cumulative Gain (NDCG)*⁴ as evaluation metric. Specifically, $NDCG@k$ is defined as follows.

$$NDCG@k = \frac{DCG@k}{iDCG@k}, \quad (21)$$

where $iDCG$ is the ideal DCG and DCG is defined as

$$DCG@k = Gain_1 + \sum_{n=2}^k \frac{Gain_n}{\log_2 n}, \quad (22)$$

where $Gain_n$ is equal to the rating of the n -th teleplay.

⁴http://en.wikipedia.org/wiki/Discounted_cumulative_gain

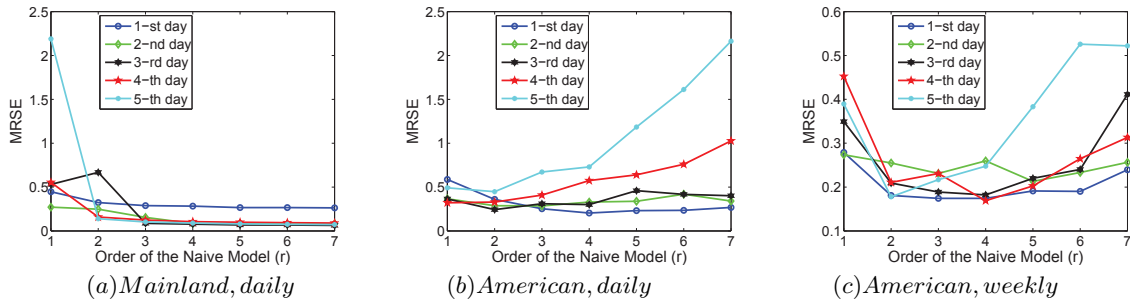


Figure 6: NAR with sharing parameters on different data sets by daily or weekly sampling.

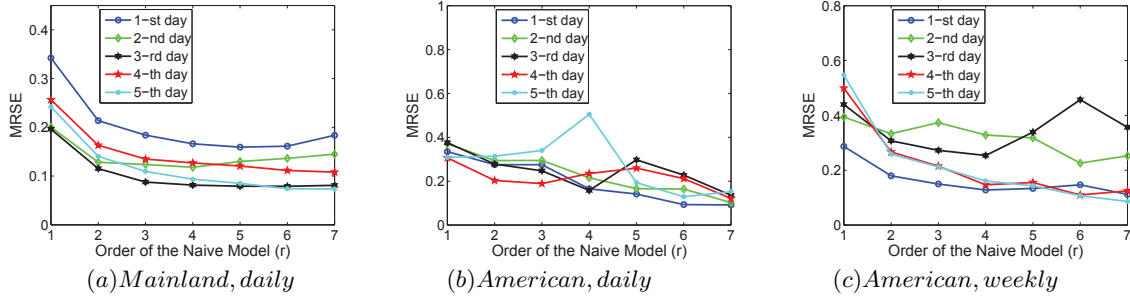


Figure 7: NAR with individual parameters on different data sets by daily or weekly sampling.

Particularly, smaller *MRSE* indicates better performance of popularity prediction while larger *NDCG* indicates better performance of quality estimation.

5.3 Performance of the NAR Model

As introduced in Section 3, NAR contains two strategies, namely sharing and individual parameters, when training the model. Therefore, we separately validate the two strategies in terms of predicting the views of the k -th ($1 \leq k \leq 5$) unit time interval after an episode is released. And the hyper-parameter λ of individual parameters is set to $\lambda = 1.6$ according to grid search. To reduce the uncertainty of model training, we utilize a 3-fold cross validation to evaluate the NAR model with sharing parameters.

Specifically, Figure 6 and 7 show the average *MRSE* of NAR with different order $r \in [1, 7]$ (i.e., using the popularity of previous r episodes) on two data sets. From these figures, we can have the following observations. First, in general, the model with individual parameters has better prediction performance than the model with sharing parameters, especially for larger r . For example, on the *Mainland* data set, the 5-order model with individual parameters for the 1-st day can lead to a 38.5% reduction of *MRSE* compared with the model with sharing parameters. Second, for *Mainland* teleplays, *MRSE* of the NAR model, both with sharing and individual parameters, decreases as the order r increases, and nearly achieves minimum value when the order $r = 3$ (i.e., from Figure 6(a) and 7(a)). Third, for *American* teleplays, *MRSE* of the NAR model with sharing parameters may increase as the order r increases (i.e., from Figure 6(b) and 6(c)), which is quite different from the results of *Mainland* teleplays. We think the possible reason of this phenomenon is that the trends of popularity for different *American* teleplays vary dramatically, thus using too much previous information would magnify the error when sharing parameters. Fourth, using a week instead of a day as unit time for model training may obtain better prediction performance, especially for the NAR model with sharing parameters. Finally, the performance of popularity prediction within different unit time intervals may be different. For example, sometimes the 1-st day can achieve the best prediction performance (e.g., Figure 6(b)), while sometimes will

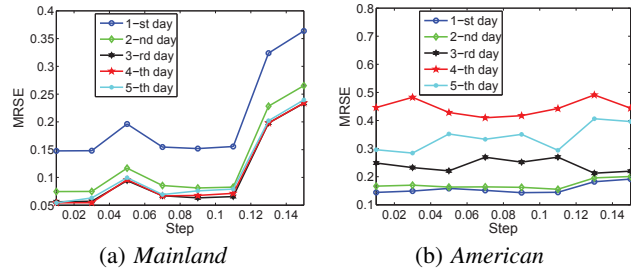


Figure 9: The effect of searching step on MRSE.

achieve the worst performance (e.g., Figure 7(a)). This observation may indicate different types of teleplays usually have different popularity change rules.

In conclusion, individual parameters and larger sampling interval can sincerely improve the performance of NAR. But using too much historical information does not mean better performance with respect to popularity prediction, and it is better to exploit sharing parameters for serials which only have released several episodes.

5.4 Performance of the STAR Model

In this subsection, we validate our single-chain transfer autoregressive (STAR) model in terms of popularity prediction and quality estimation on the two data sets.

5.4.1 Popularity Prediction

As discussed in Section 4.1, here we evaluate the STAR model with different settings of model parameters p and q , i.e., 1) fixing $p \& q$, 2) increasing $p \& q$, and 3) decreasing $p \& q$. Particularly, for training the STAR model, we assume that views of the first $N = 8$ episodes have already been known⁵. Figure 8 shows the prediction performance of the STAR model with respect to different unit time intervals after the episodes are released. From these results, we can first find that *MRSE* of STAR with both fixing and increasing $p \& q$ have a similar trend, which are different with the performance of

⁵Here $N = 8$ is a trade-off for model training, since our pre-experiments show that the models trained with $N \geq 8$ have very similar parameters.

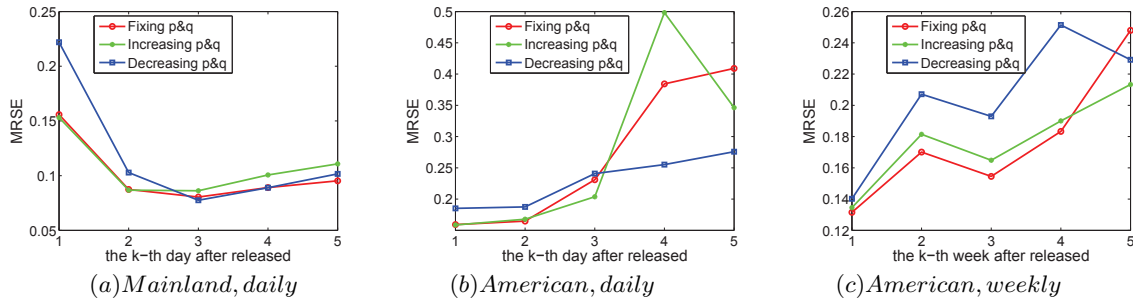


Figure 8: Prediction performance of the STAR model on different data set by daily or weekly sampling.

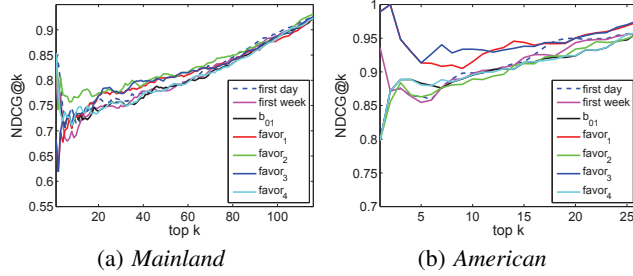


Figure 11: NDCG@k on different data set.

the STAR model with decreasing $p&q$. Second, the prediction performance of STAR model on *Mainland* teleplays is different with the performance on *American* teleplays. At last, using a week instead of a day as unit time for model training may obtain better prediction performance, which is similar to the results of STAR.

In Algorithm 1, we have a parameter, namely searching *step*, for learning the appropriate parameters by grid search. Here, we evaluate the prediction performance of STAR with different settings of *step* when fixing $p&q$ in the algorithm, and the results are shown in Figure 9. We can observe that the *MRSE* performance has an increasing trend when the *step* gets larger. However, larger *step* can reduce the running time of Algorithm 1. Therefore, $step \approx 0.1$ is a good trade-off between efficiency and effectiveness.

As a summary, we believe the STAR model with fixing $p&q$ is the most appropriate model, since it is both effective and simple. The above also proves that popularity change rules of different types of teleplays are very different.

5.4.2 Quality Estimation

For quality estimation, we use the ratings from *Douban*⁶ as the ground truth to measure the quality of teleplays. Specifically, larger rating indicates higher quality. We first filtered out teleplays without *Douban Rating* in our data sets. The final number of teleplays in *Mainland* and *American* data sets are 116 and 26, respectively. We sort the teleplays by ratings and calculate the *iNDCG*. For each teleplay, views of all episodes on the first day are used to train the STAR (i.e., with fixing $p&q$) and different *favor* metrics can be computed with the learned parameters. To validate the effectiveness of our novel metric *favor*, we compare it with two popularity based metrics, namely views of the first day and views of the first week of each episode. Also, we use the model parameter b_{01} , which represents the initial public attention of the teleplay, as the third baseline. Figure 11 shows the results of quality estimation on different data sets. From the results, we can observe that the *favor* metrics have better *NDCG@k* performance than other baselines, especially at small k . Particularly, *favor*₂ has the highest

NDCG@k performance for almost every k on the *Mainland* data set, while *favor*₁ and *favor*₃ have the highest *NDCG@k* performances on the *American* data set. Therefore, we can claim that the *favor* is effective for estimating the quality of online serials.

5.5 Comparison of Models

In this section, we compare our prediction models with baselines, the *Average* method and 1-VAR. Specifically, by carrying out experiments on two data sets, we find that the *Average* method can achieve the best prediction performance when using average views of previous 4 episodes. Therefore, here we use the 4-order *Average* method in comparison, which is denoted as 4-Average. Meanwhile, for 1-VAR, we set the dimensionality of each vector to be 5, which can achieve the best prediction performance compared with other dimensionalities. The settings of our models are the 3-NAR model with sharing parameters (3-NARSP), 3-NAR model with individual parameters (3-NARIP) (i.e., the prefix 3 here denotes the order of the naive autoregressive model) and STAR model with fixing $p&q$. The detailed comparison results are demonstrated in Figure 10.

From the results, we can first find that the performance of 4-average and 1-VAR are very unstable and have the highest *MRSE*. The results may indicate that directly extending VAR and using the *Average* method to predict the popularity of online serials are not proper. Second, our NAR and STAR model are not sensitive to the settings of unit time interval k , and using a week instead of a day as unit time for model training may obtain better prediction performance. Third, the strategy of individual parameters can improve the performance of the NAR model. Finally, the STAR model outperforms the NAR model in terms of popularity prediction.

5.6 Case Study

In order to better understand the performance of our methods in terms of popularity prediction for serials, we conduct some case studies on the *American* data set.

Here, we run the 3-NAR model with sharing parameters (3-NARSP), 3-NAR model with individual parameters (3-NARIP) and STAR model with fixing $p&q$ on a specific *American* teleplay, namely *Zero Hour Season 1*, respectively. Specifically, Figure 12 shows the prediction results of different episodes on different days after they are released. From the results, we can have the following observations. First, for small episode IDs, the relative errors of the NAR model with individual parameters are larger than the NAR model with sharing parameters. It is because that there are not enough episodes for learning the individual parameters. Second, the STAR model can achieve a significant improvement over the NAR model in terms of popularity prediction, especially for the first few days. For example, in Figure 12(a), the relative error of the STAR model of the 9-th episode on the 1-st day after it was released is 0.055, while the other two corresponding relative errors of the NAR model are 0.262 and 0.419.

⁶<http://movie.douban.com/>

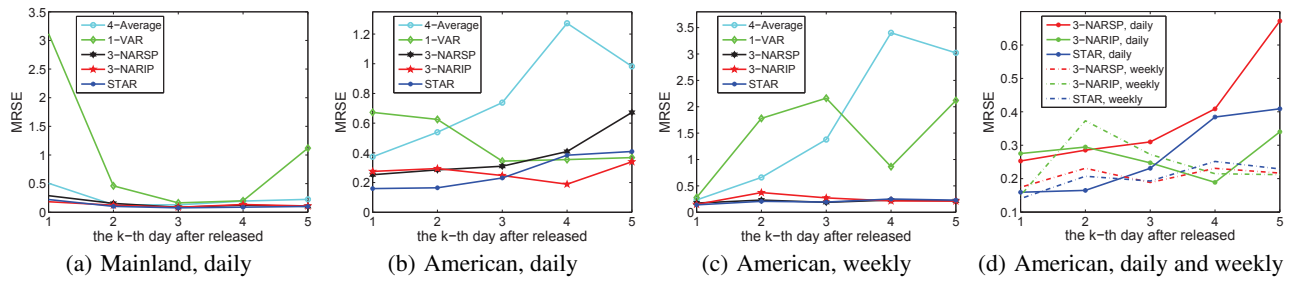


Figure 10: Comparison of models on different data sets by daily or weekly sampling.

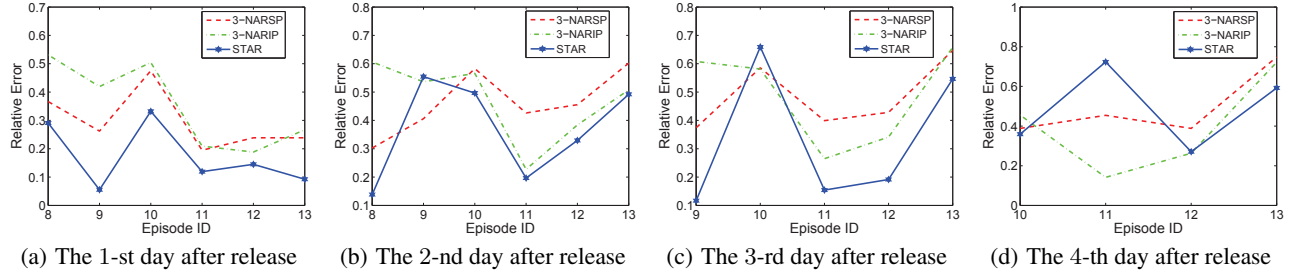


Figure 12: Popularity prediction of different episodes on the k -th ($1 \leq k \leq 4$) day after release for a specific American teleplay.

6. RELATED WORK

Generally, the related works of this paper can be grouped into two categories, namely popularity prediction for online content and time series forecasting.

Popularity Prediction. There have been a lot of efforts devoted to popularity analysis and prediction for online contents. For example, [7] analyzed the popularity distribution, popularity evolution, and content duplication of YouTube videos. Particularly, they found that leveraging information filtering can increase the total views of nice contents as much as 45% and caching the most popular contents can offload server traffic by 50%. [8] studied the relaxation response of a social crowd after endogenous and exogenous bursts of activity and clustered the patterns into four main classes, namely *memoryless*, *viral*, *quality* and *junk*. Besides, [10] characterized the growth patterns of video popularity on YouTube and the types of the referrers that most often attracted users to each video. [5] developed a methodology which can accurately assess the impacts of various content-agnostic factors on video popularity. They found a strong linear “rich-get-richer” behavior with the total number of previous views when controlling for video content.

On the other hand, [21] discovered that there is a correlation between early votes and future popularity of online contents up to a normally distributed noise. They used a logarithm linear model to fit the correlation. [20] extended their work [21] and proposed the *multivariate linear model* and *multivariate radial basis function model*. The former uses the number of views received per sampling interval as features, instead of the total number of early views. The latter extracts more features by utilizing radial basis functions to measure the similarity between different videos. Furthermore, [1] first designed two new types of features, i.e., *SA* (capturing the *Share of Attention* of videos) and *nROC* (capturing the *normalized Rate of Change in the attention attracted*). Then they clustered views in each sampling interval according to the above features and took advantage of the maximum likelihood path to predict the popularity of a given content. While [29] exploited the latent conforming and maverick personalities to develop a conformer-maverick model and used it to rank top- k potentially popular items based on the early votes they received.

In addition, [18] constructed both content and contextual features for hashtag popularity prediction from tweets containing the hashtag. Whereas [28] presented a novel task of Citation Count Prediction given papers’ citation patterns (such as paper content, author expertise and venue impact) and discovered that authors have biases in citing references.

Time Series Forecasting. Models for time series forecasting have many forms and can describe different stochastic processes. For example, the autoregressive model (AR) [25] can describe the process of a single time series. It assumes that the current value linearly depends on its own previous values. While vector autoregression (VAR) models [17] generalize AR and can capture the linear interdependencies among multiple time series. “A VAR model describes the evolution of a set of k variables (called endogenous variables) over the same sample period ($t = 1, \dots, T$) as a linear function of only their past values” [26]. Furthermore, [3] addressed how to include exogenous variables in a VAR model, and what the consequences are for the innovation response analysis. In particular, [12] exploited the VAR models to forecast house prices for the four census regions and the aggregate US economy in a data-rich environment.

However, these methods cannot work well in our scenario. Different from previous works, we are interested in predicting the popularity of online serials. They usually contain several regularly updated episodes, which have unique characteristics of sequence dependence, release date dependence as well as the unsynchronized update regularity. Different release dates lead to that vectors of popularity at different time have different dimensionalities, which restrains the utility of techniques in time series forecasting. Therefore, in this paper we propose two novel autoregressive models, namely the Naive Autoregressive (NAR) model and Transfer Autoregressive (TAR) model, to capture these characteristics and predict the popularity of online serials.

7. CONCLUSION AND FUTURE WORK

In this paper we studied the problem of predicting the popularity of online serials by leveraging autoregressive models. We developed two novel models, namely *Naive Autoregressive* (NAR) and *Transfer Autoregressive* (TAR) model, which can reveal the

popularity generation of online serials from different perspectives. Specifically, the NAR model is based on the episode correlations of serials, which is straightforward but effective, and the TAR model can capture the dynamic behaviors of audiences. As a derivative of the TAR model, we also designed a novel metric, namely *favor*, for evaluating the quality of online serials. Finally, the experimental results on two real-world data sets clearly demonstrated the effectiveness of the proposed models and metric.

In the future, we will study more possible solutions of the TAR model. Moreover, we plan to integrate some external information into our models for popularity prediction, such as search volume, rating information and user reviews from social networks.

8. ACKNOWLEDGEMENT

This work was supported in part by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010) and the National High Technology Research and Development Program of China (Grant No. 2014AA015203). This work was also partially supported by grants from the National Science Foundation (NSF) via grant numbers CCF-1018151 and IIS-1256016, the Anhui Provincial Natural Science Foundation (Grant No. 1408085QF110).

9. REFERENCES

- [1] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini. A peek into the future: predicting the evolution of popularity in user generated content. In *WSDM'13*, pages 607–616. ACM, 2013.
- [2] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. Ramakrishnan. Optimal content placement for a large-scale vod system. In *CoNEXT'10*, page 4. ACM, 2010.
- [3] H. J. Bierens. Var models with exogenous variables. 2004. Available on http://econ.la.psu.edu/hbierens/EasyRegTours/VAR_Tourfiles/VARX.PDF.
- [4] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.
- [5] Y. Borghol, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti. The untold story of the clones: content-agnostic factors that impact youtube video popularity. In *SIGKDD'12*, pages 1186–1194. ACM, 2012.
- [6] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [7] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC'07*, pages 1–14. ACM, 2007.
- [8] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *NAS*, 105(41):15649–15653, 2008.
- [9] J. Famaey, T. Wauters, and F. De Turck. On the merits of popularity prediction in multimedia content caching. In *IM'11*, pages 17–24. IEEE, 2011.
- [10] F. Figueiredo, F. Benevenuto, and J. M. Almeida. The tube over time: characterizing popularity growth of youtube videos. In *WSDM'11*, pages 745–754. ACM, 2011.
- [11] Y. Ge, G. Jiang, and Y. Ge. Efficient invariant search for distributed information systems. In *ICDM'13*, pages 1049–1054, 2013.
- [12] R. Gupta. Forecasting house prices for the four census regions and the aggregate us economy in a data-rich environment. *Applied Economics*, 45(33):4677–4697, 2013.
- [13] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW'10*, pages 591–600. ACM, 2010.
- [14] H. Lakkaraju and J. Ajmera. Attention prediction on social media brand pages. In *CIKM*, pages 2157–2160. ACM, 2011.
- [15] K. Lerman and T. Hogg. Using a model of social dynamics to predict popularity of news. In *WWW'10*, pages 621–630. ACM, 2010.
- [16] C. Li and C. Chen. Measurement-based study on the relation between users' watching behavior and network sharing in p2p vod systems. *Computer Networks*, 54(1):13–27, 2010.
- [17] H. Lütkepohl. *New introduction to multiple time series analysis*. Springer, 2007.
- [18] Z. Ma, A. Sun, and G. Cong. Will this# hashtag be popular tomorrow? In *SIGIR'12*, pages 1173–1174. ACM, 2012.
- [19] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.
- [20] H. Pinto, J. M. Almeida, and M. A. Gonçalves. Using early view patterns to predict the popularity of youtube videos. In *WSDM'13*, pages 365–374. ACM, 2013.
- [21] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.
- [22] A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida. Predicting the popularity of online articles based on user comments. In *WIMS*, page 67. ACM, 2011.
- [23] E. Veach. Using enhanced ad features to increase competition in online advertising, Feb. 27 2014. US Patent 20,140,058,829.
- [24] P. Whittle. *Hypothesis testing in time series analysis*, volume 4. Almqvist & Wiksells, 1951.
- [25] Wikipedia. http://en.wikipedia.org/wiki/autoregressive_model. 2014. [Online; accessed 14-January-2014].
- [26] Wikipedia. http://en.wikipedia.org/wiki/vector_autoregression. 2014. [Online; accessed 10-January-2014].
- [27] T. Wu, M. Timmers, D. D. Vleeschauwer, and W. V. Leekwijck. On the use of reservoir computing in popularity prediction. In *INTERNET'10*, pages 19–24. IEEE, 2010.
- [28] R. Yan, J. Tang, X. Liu, D. Shan, and X. Li. Citation count prediction: learning to estimate future citations for literature. In *CIKM'11*, pages 1247–1252. ACM, 2011.
- [29] P. Yin, P. Luo, M. Wang, and W.-C. Lee. A straw shows which way the wind blows: ranking potentially popular items from early votes. In *WSDM'12*, pages 623–632. ACM, 2012.
- [30] B. Yu, M. Chen, and L. Kwok. Toward predicting popularity of social marketing messages. In *SBP'11*, pages 317–324. Springer, 2011.
- [31] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. In *ACM SIGOPS Operating Systems Review*, volume 40, pages 333–344. ACM, 2006.
- [32] H. Zhu, E. Chen, K. Yu, H. Cao, H. Xiong, and J. Tian. Mining personal context-aware preferences for mobile users. In *ICDM'12*, pages 1212–1217, 2012.
- [33] H. Zhu, H. Xiong, Y. Ge, and E. Chen. Mobile app recommendations with security and privacy awareness. In *SIGKDD'14*, 2014.